Experiences Developing Distributed Object Applications

David Forslund

Los Alamos National Laboratory

OOPSLA'98

October 22, 1998

Los Alamos TeleMed

PROBLEM:

- A significant percentage of people have long-term chronic illnesses treated in multiple locations and times
- Provide a secure, lifetime longitudinal electronic medical record for any person on the planet anywhere they are without requiring a central repository.
- A legitimate GRAND CHALLENGE.
- Potential of significantly improving the quality of healthcare worldwide. (80 million will die from TB in the next 5 years!)

GOAL:

- Build a virtual medical record from an arbitrary set of locations which must be dynamically discovered. (more than the web!)
- Access must be negotiated at runtime
- We have built a prototype of such a system and are working on a process of reaching the ultimate goal.

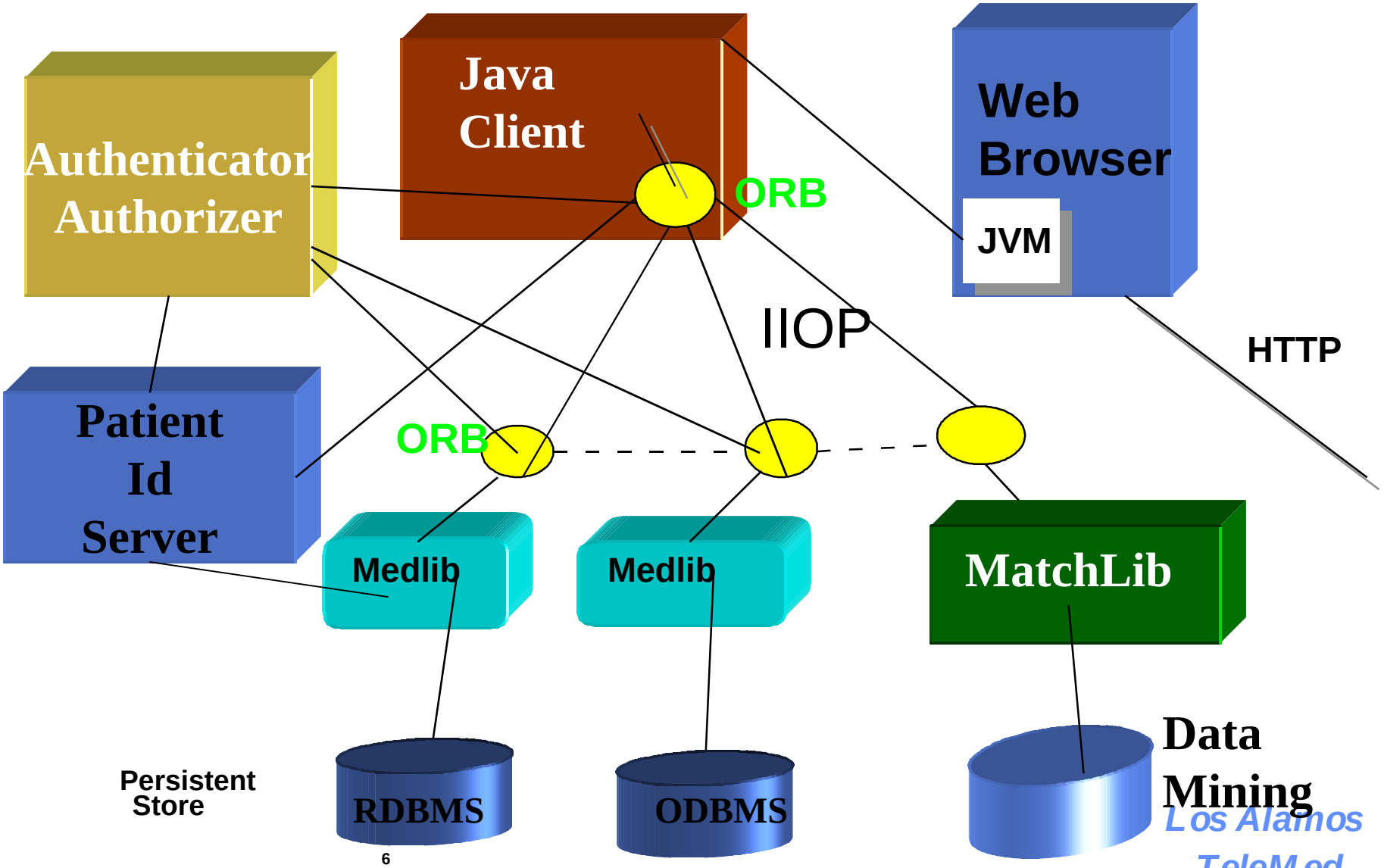
APPROACH:

- **ISSUES**
  - **Everyone use the same hardware and software implementations?**
  - **NO!**
  - **There is no agreement on the content of an EMR.**
- **Build a set of components based on standard interfaces which allows the medical record to be dynamically constructed and evolved.**
- **CORBA was chosen because of its language, platform, and binary independence and has been available for more than 5 years**

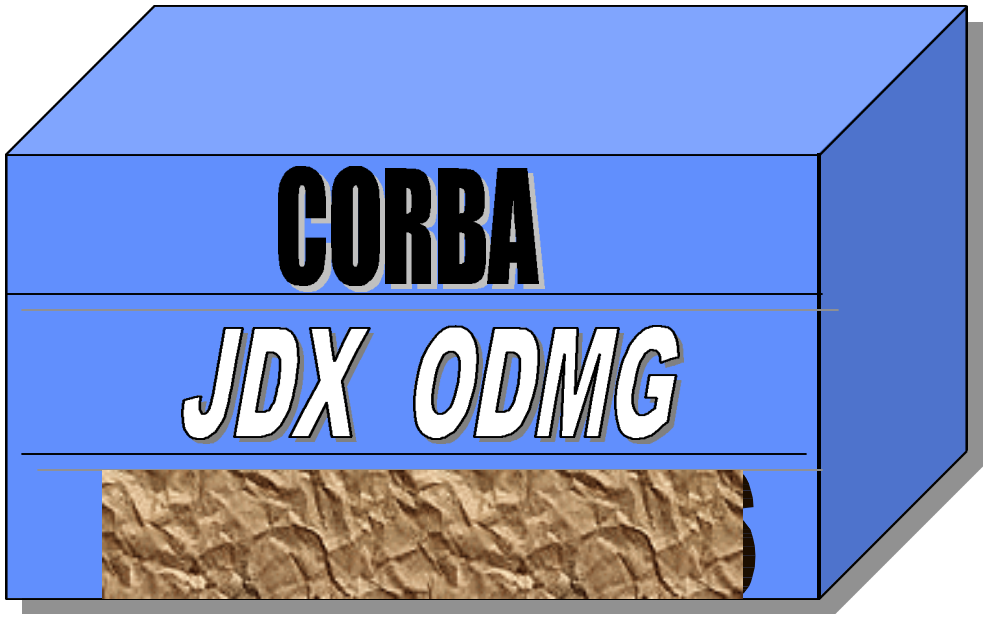
APPROACH...

- Standards already exist for many of the data elements (HL7, XML/EDI, ...), but these are not objects
- We wanted to use objects to provide encapsulation, polymorphism, ease of management.
- CORBAmed is creating the standard interface specifications
- We have built a prototype totally in Java for increased portability.
- We are in our 4th generation of the application.
  - GainMomentum/CORBA/ObjectStore: 94-95
  - Java/CORBA/ObjectStore 96
  - Java/CORBA/Java 97
  - Java/CORBA/Java 98 (OMG PIDS compliant)

TeleMed Architecture



Server architecture



TeleMed Principles

- Move data only as necessary
- Manage complex high-volume data in understandable manner
- Same system for real-time consultations as for asynchronous consultations
- Leverage and enhance internet standards (e.g., OMG)
- Design for intuitive ease of use in clinical setting
- Plug-and-play design
- Have at least two vendors at every level

EXPERIENCES

- Political issues are much larger than software issues
- Security policy controls deployment, not the software
- FDA software approval is an issue (what does this mean worldwide?)
- Java has helped us develop much faster and cleaner than we were able to do in C++. JVM versioning has been a problem.
- During the evolution of the system, we reduced the use of inheritance and moved toward compositional models

Java™ Experiences

- **Development has been much faster and cleaner**
  - **no management of macros issues!**
  - **Easier memory management**
- **Server side has been very successful**
  - **ODMG Java bindings work well for us**
- **JVM version transition (1.0.2-1.1) was painful mostly on the client**

CORBA Experiences

- CORBA works well, but has room for improvement (e.g., more rigorous specifications, objects by value)
- Using CORBA is complicated but then the management of the wide area dynamic distributed computing environment isn't trivial
- CORBA services have been slower coming than we anticipated
- CORBA isn't modular enough, yet. Too vendor focused.
- Convergence between CORBA and the Web

Summary

- A Worldwide distributed EMR is achievable in our opinion, but needs to be completely dynamic and extensible.
- Social and political issues will limit the deployment of such a system
- Objects are slowly coming to the web and we see a strong convergence between CORBA/XML/Java occurring